Fine-tuning Regression Forests Votes for Object Alignment in the Wild

Heng Yang, *Student Member, IEEE,* Ioannis Patras, *Senior Member, IEEE*

***Abstract*—In this paper we propose a object alignment method that detects the landmarks of an object in 2D images. In the Regression Forests (RF) framework, observations (patches) that are extracted at several image locations cast votes for the localization of several landmarks. We propose to refine the votes before accumulating them into the Hough space, by sieving and/or aggregating. In order to filter out false positive votes, we pass them through several sieves, each associated with a discrete or continuous latent variable. The sieves filter out votes that are not consistent with the latent variable in question, something that implicitly enforces global constraints. In order to aggregate the votes when necessary, we adjusts on-the-fly a proximity threshold by applying a classifier on middle-level features extracted from voting maps for the object landmark in question. Moreover, our method is able to predict the unreliability of an individual object landmark. This information can be useful for subsequent object analysis like object recognition. Our contributions are validated for two object alignment tasks, face alignment and car alignment, on datasets with challenging images collected *in the wild*, i.e. the Labeled Face in the Wild, the Annotated Facial Landmarks in the Wild and the street scene car dataset. We show that with the proposed approach, and without explicitly introducing shape models, we obtain performance superior or close to the state of the art for both tasks.**

***Index Terms*—Regression Forests, facial feature, object alignment.**

I. Introduction

DEFORMABLE object alignment in an image is one of the most important and well studied problems in computer vision, where the shape of the object, such as face or car, is typically described by a set of landmarks $\mathcal{S} = \{h_1, ..., h_j, ..., h_J\}$, with h_j the coordinates of the jth landmark in the shape space. This alignment step is very crucial for a variety of applications like face recognition, object reconstruction and tracking. In the past decades, it has been studied extensively and many alignment models have been proposed, particularly for the face alignment, for instance the Active Appearance Model (AAM) [1]. Recent approaches on object alignment attempt to make a transition from images recorded in controlled conditions to images "in the wild", for instance [2], [3], [4], [5], [6] for face alignment and [7] for car alignment. However, they still have difficulties with low quality face images, object pose variations and partial occlusions, especially when high accuracy and real-time detection are needed.

H. Yang and I. Patras are with the School of Electrical Engineering and Computer Science, Queen Mary University of London, London, UK, E1 4NS. e-mail: {heng.yang,i.patras}@qmul.ac.uk.

In this paper, we address the problem using random decision forests, given their good performance in various challenging computer vision tasks like action recognition [8], human pose estimation [9], head pose estimation [10] and facial feature detection [3]. We use random Regression Forest in this work, which is also regarded as an important instance of the Generalized Hough Transform [8], [11]. In this framework, image patches that are densely extracted from the image and are propagated through each tree of the forest until they arrive at leaf nodes, where, they cast votes in the Hough space of the parameters to be estimated. In general, the number of votes is very large since the patches are densely extracted from the whole image or region of interest. Taking all the voting elements into account can lead to a bias towards the mean shape, i.e. the average shape of a large number of samples. Therefore, it is common practice in Regression Forests set a threshold that does not allow elements with large offsets to vote, e.g., [3], [9]. This threshold controls the maximum allowed offsets of votes. We in this paper transform the Euclidean distance d into a proximity metric by a function $f(d) = e^{-d}$. Therefore longer distance corresponds to lower proximity. We then equivalently set a threshold on the proximity on the lower bound. If a large threshold is set on proximity, only votes with small offsets are allowed. Those are expected to have high localization accuracy, unless they are contaminated due to conditions like noise, shadows and occlusions. On the contrary, with a small threshold on proximity, votes are allowed from observations that are far and in this way introduce shape constraints and robustness to occlusions. The extreme case of the latter situation is using all voting elements from the region of interest. Typically, such a threshold is optimized at training stage and kept fixed during testing.

There are two issues with the Regression Forest mechanism described above. First, different landmarks are voted in a completely independent way, i.e. there is no mechanism that encourages consistent predictions. Therefore we might see implausible detection results that violates the shape consistency. Second, keeping the proximity threshold fixed creates problems on challenging images, for example when some landmarks are partially occluded. In extreme cases, the size of the occluded area is big but very high proximity threshold is set. In such cases, very few 'good' votes can be obtained, which in turn results in localization failures. To address these two problems, we propose a method that refines the votes, by sieving and/or aggregating them, before they are accumulated into the Hough space, as described below.

Votes Sieving introduces a bank of sieves, each one as-

sociated with one of a few latent variables such as the head pose discrete label, or the location of some anchor points for instance the object center. Essentially, each sieve operates as a filter that rejects votes that are not consistent with hypotheses of the latent variable in question. This introduces global consistency and effectively eliminates the false positive votes.

Votes Aggregating learns a model that for each test image, decides whether or not to reduce the proximity threshold for an individual object landmark. This is opposed to using a fixed proximity threshold and controls the extend of spatial constraints. For example, with a low proximity threshold for the localisation of the left-eyebrow corner, one would also use the votes of test-patches that arrive at leaf nodes to which training patches that were extracted close to other facial areas as well (e.g. left eye, or even nose), mostly arrive. Since test patches that vote for the location of more that one landmark introduce shape constraints between the location of those landmarks, the proximity threshold can be seen as controlling the spatial extend of the (implicit) shape model that we use. The decision is made based on a classifier that is built on middle-level features which are extracted from the current votes for the location of the landmark in question.

Another contribution is that our approach predicts the unreliability of each individual landmark. The unreliability tells the probability of the landmark is with occlusion or other clutter and provides useful information for subsequent high-level object analysis for instance face recognition.

The experimental results show that without using typical shape constraints, the approach achieves results superior or comparable to state-of-the-art on two challenging face landmarks datasets, namely, the Labelled Faces in the Wild, the Annotated Facial Landmarks in the Wild. In addition, since no specific model is assumed, the same approach can be applied without modification to other domains. This is illustrated by applying the approach for car alignment and we obtain the state-of-the-art results. We also show that the benefits of using sieves and determining an image-dependent and landmark-dependent threshold are higher for the 'difficult' images in both datasets.

The rest of the paper is organised as follows. Related work on random forests and object alignment are given in Section 2. In Section 3 we introduce the proposed method. Experimental results and comparisons with the current state-of-the-art methods are given in Section 4 and in Section 5 we draw some conclusions.

II. Related work

In this section we first present an overview of related work on object landmarks detection, and then present a brief review of the random forests literature that is relevant to this work.

A. Object Alignment

Object alignment is a well-studied problem in computer vision, especially for the face. Two different sources of information are typically exploited for this task: appearance and spatial shape. Based on how those two types of information are utilized, we categorize the methods into three groups: discriminative local landmark detection, part-based deformable models and explicit shape regression.

Discriminative Local Detection approaches only exploit the discriminative appearance features of different landmarks. A regressor or a classifier is learned independently for each landmark. For instance, in [12], GentleBoost classifier based on Gabor features is proposed to detect 20 facial points separately. The Support Vector Machine (SVM) classifier is used as facial point detector in [13] and [2] . Regression Forests (RF) are introduced as a local detector for facial landmarks [3] and human body joints [9] detection. In [14], Boddeti *et al.* introduced Correlation Filters to learn a local appearance model. Methods in this category can be regarded as an extension of general object detection. Since no shape constraints are imposed, this type of methods have good generality but suffer heavily from partial occlusions.

Part-based Deformable Models focus on using shape prior to regularize the local part detections. Thus the two tasks at training time are first learning the part models and second learning the shape prior. The part model is often learned following the discriminative local detection methodology described above. Typical shape models like the Constrained Local Models (CLMs) [15] first align the images using the landmarks annotation by Procrustes analysis, then learn the shape prior by using Principal Component Analysis (PCA). Other shape models include the probabilistic MRF model in [16] and tree structured shape models proposed by Zhu and Ramanan [5]. The latter has shown good results both in capturing global elastic deformation and in finding the global optimal solutions by linear programming. Amberg et al. [17] proposed to find an optimal set of local detections using a Branch & Bound method. Recently, several methods proposed non-parametric shape constraints, for example, the RANSAC-based methods in [2] and [7]; the regularized mean-shift model in [18]; the graph-matching method in [19] and shape constraints within the regression forests in [20], [21].

Explicit Shape Regression approaches jointly model the shape and appearance, and learn directly a mapping from image features to the shape space (the location of the landmarks). A typical method is this category is the Active Appearance Model (AAM) proposed by Cootes *et al.* [1]. The AAM is fit by learning a linear regression between the appearance differences and the increment of alignment parameters. Since a very simple linear regression method is applied, the original fitting method suffers from occlusion and is very difficult to deal with unseen images. Saragih and Gocke [22] and Tresadern *et al.* [23] showed that using boosted regression for AAM discriminative fitting significantly improved the original linear formulation. In the boosting/cascade framework, recent methods [24], [4], [6] learn a set of weak regressors (random ferns) to model the relation between the image feature and the update of the parameters. They introduce a cascade in which they re-sample features based on the current shape state and use them in the next regressor. Xiong and De la Torre [25] proposed a similar cascaded method and used more advanced image features (HoG [26] and SIFT [27]) that improves the performance.

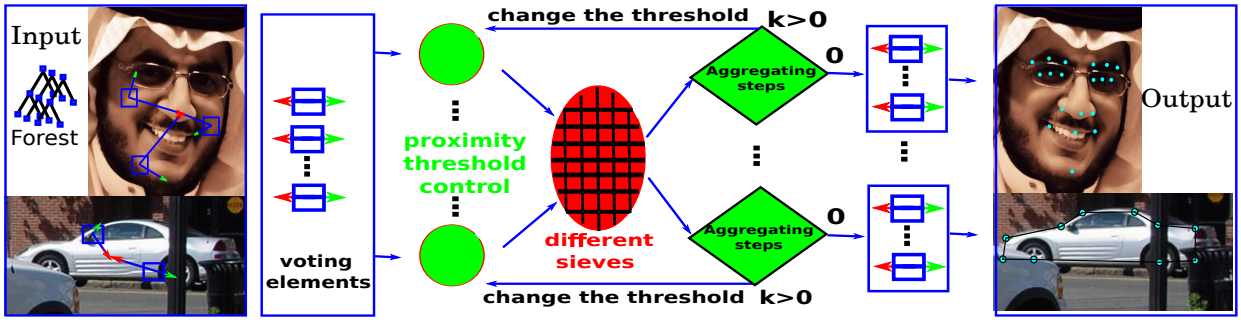


Fig. 1: Framework of the method.

B. Random Forests

Random forests have emerged as a powerful and versatile method that has been successful in real-time human pose estimation, semantic segmentation, object detection and action recognition [8], [10], [28]. A comprehensive introduction of decision forests and their applications in computer vision is given in [29]. We in this paper build the random Regression Forest that is similar to the framework used for facial feature detection [3], 3D head pose estimation [30] and Human Pose Estimation (body joints prediction) [28].

A regression forest is an ensemble of decision trees that predicts continuous outputs. Each binary tree consists of internal nodes and leaf nodes. The internal nodes contain test functions that evaluate the input features to decide whether to go to the left or to the right child nodes. The terminal nodes contain continuous prediction models, as opposed to the categorical prediction in classification forests. At the training time, a set of image patches are randomly extracted from the training images. Each of them contain the offset information, e.g. the patch center to each of the facial landmarks. At each internal node, a pool of candidate functions is randomly generated. The one that maximizes the information gain is selected as the split function at that nodes. This process is recursively applied until a certain stopping criterion is met, such as that the maximum depth of the tree is reached or the number of patches is less than a threshold. During testing time, image patches are densely extracted from the image and fed to each tree. When one patch arrives at a leaf node, there is a regression model, usually a relative offset vector (i.e. vote) to each of the landmarks of interest (potentially to all landmarks), along with a weight. In this way, a regression forest transforms the image observation to a set of votes.

Some recent Regression Forests introduce latent variables and use the additional information at the training phase. Sun *et al.* [9] propose a conditional regression forest model for human pose estimation. During training, at each leaf node, the vote is decomposed into the distribution of 3D body joint locations for each codeword at the leaf node and the codeword mapping probability. In their model, they use the latent variables like the body height and body orientation. They propose a method to jointly estimate the body joint locations and the latent variable. Dantone *et al.* [3] also introduced a regression forests model conditioned on latent variable of head pose for face alignment. In their method, they divide the training set into subsets according to head pose yaw angles (left profile, left, front, right, right profile). An individual regression forest is trained on each subset. During testing a set of regression trees is selected according to the estimated probability of the head pose yaw angle. The later is given by an additional forest trained to perform head pose estimation.

Our approach is also closely related to those methods that analyse votes returned from regression forests. For instance, [31] modelled the joint distribution over all the votes and the hypotheses in a probabilistic way, rather than simply accumulating the votes. [11] studied the geometric compatibilities of the voting elements in a pairwise fashion within a game-theoretic setting. These methods are developed for person detection and focus on *intra-class* geometrical agreement, while the consistency in our problem is *inter-class*, since we consider the localization of multiple landmarks on an object. In terms of rejecting irrelevant observations for regression, our work is related to [32] that applies classifiers for observation selection for robust object tracking.

III. Method

In this section, we describe the proposed method as answering the problem of face alignment/facial feature detection. We first introduce the latent space and the votes from regression forests associated with the latent variable. Then we describe the votes sieving strategy based on the latent variable agreement and finally we describe the votes aggregating and landmark unreliability detection.

A. RF Votes with Latent Variable

For tree construction, we follow the procedure proposed in [3]. We use the information gain (IG) as the criterion to select the split function. A entropy-like class uncertainty $\mathcal{H}(\mathcal{I})$ on a set of image patches $\mathcal{I}$ is defined as:

$$\mathcal{H}(\mathcal{I}) = -\sum_{h_j \in H} \frac{\sum_{I_i \in \mathcal{I}} p(h_j|I_i)}{|\mathcal{I}|} \log\left(\frac{\sum_{I_i \in \mathcal{I}} p(h_j|I_i)}{|\mathcal{I}|}\right), \tag{1}$$

$$p(h_j|I_i) \propto f(|d_{h_j}^{I_i}|) = \exp\left(-\frac{|d_{h_j}^{I_i}|}{\alpha}\right), \tag{2}$$

where $p(h_j|I_i)$ indicates the probability that the patch I_i belongs to the the j-th landmark [3], $j \in 1, ..., J$. We use h_j denote the location of the landmark. $f(\cdot)$ is a function that transforms the Euclidean distance $d_{h_j}^{I_i}$ into a proximity metric. This proximity metric is used throughout this paper.

The constant α controls the steepness of this function Note that the distance measure d is normalized by the object size.

Once the regression forest is trained, the observations, i.e. image patches $I_i \in \mathcal{I}$ are extracted from the testing image location y_i and fed to it. When they arrive at leaf nodes they cast weighted votes $v(h|I_i)$ for the location of one or more landmarks. For a given hypothesis $h \in H$, the score of h is determined by the sum of votes that support the hypothesis: $S(h) = \sum_i v(h|I_i)$. In practice each patch I_i will be sent to each tree $t \in T$ in the forest, i.e. $S(h) = \sum_i \sum_t v(h|I_{it})$. We will drop the t in the subsequent discussion for clarity and consistency with other methods.

With the procedure described above, there are some votes are inconsistent with some latent variables, for instance, in head pose and face center. These votes are unlikely to vote correctly. Some previous work [33] proposes to augment the hypothesis space by a latent space Z to enforce consistency of the votes in some latent properties $z \in Z$. That method can only deal with discrete latent variables and has high memory requirements and computational complexity, when large training data is used since all training patches need to be stored. By contrast, the latent space in our work can be either discrete or continuous. The score of a hypothesis in the augmented space is then given by:

$$S(h) = \sum_{i, z \in \phi(\hat{z})} v(h, z|I_i) \tag{3}$$

where $\phi(\hat{z})$ is an affiliation term defined as follows. When the latent space is discrete, $\phi(\hat{z}) = \{\hat{z}\}$, with $\hat{z}$ a discrete label. It means votes with the same latent variable as $\hat{z}$ are used. When the latent space is continuous, $\phi(\hat{z}) = \{z : ||z - \hat{z}|| \leq r\}$, where r is the radius of a region around $\hat{z}$. The details are described in Section III-B.

B. RF Votes Sieving

1) Sieving via Discrete Latent Variable: Our method of sieving votes using discrete latent variable is similar to the conditional regression forest **Partial Model** proposed in [9] that was used for human pose estimation. During training, each patch extracted from the training samples is annotated with a discrete latent label. We use the tree construction procedure proposed in III-A. When the training patches arrive the leaf node l, we learn one model for each state of the latent variable. More specifically, we first partition the training patches according to their latent variable labels and then learn a model in each partition with latent label z for the hypothesis h. The model vector is $(\Delta_l^z, \omega_l^z, p_l^z)$. where Δ_l^z is the relative offset vector, obtained by taking the center of the largest mode found by mean-shift clustering method [34] in the partition with latent label z, similar to [9]. ω_l^z is weight, given by the relative size of the largest cluster. For the latent variable z, p_l^z is the probability of the latent variable at leaf node l, that is calculated as the proportion of the training samples whose label is z, that is,

$$p_l^z = \frac{n_l^z}{\sum_{z \in Z} n_l^z} \tag{4}$$

where n_l^z is the number of training patches with the latent label z. When a patch I_i extracted from the location y_i arrives this leaf node l, the vote is represented as:

$$v(h, z|l) = \omega_l^z \delta(\Delta_l^z + y_i - h). \tag{5}$$

Since the probability of the latent variable is independent of the hypothesis, its scoring function is:

$$S(z) = \sum_i v(z|I_i) = \sum_i \sum_{h \in H} v(h, z|I_i) = \sum_l p_l^z. \tag{6}$$

The latent variable is estimated as $\hat{z} = \arg\max_{z \in Z} S(z)$. Given the estimation, the hypothesis scoring function is formed by using the votes with the corresponding latent state, i.e.,

$$S(h) = \sum_{i, z \in \phi(\hat{z})} v(h, z|I_i) \tag{7}$$

2) Sieving via Continuous Latent Variable: The tree construction process of the sieving via continuous latent variable is similar. Each training patch is associated with a continuous latent variable label, for instance the displacement to the object center. This latent information is not used until the patches arrive the leaf node. We use the face center as an example to show how continuous latent variable is modelled. The leaf model vector is $(\Delta_l^z, \omega_l^z, \Delta_l^{cz}, \omega_l^{cz})$. In addition to Δ_l^z and ω_l^z, we have two similar terms, Δ_l^{cz} and ω_l^{cz}, that are the offsets to the face center and the corresponding weight respectively, learned in a similar way of learning Δ_l^z and ω_l^z.

During testing, we will first estimate the state of the latent variable z, i.e. the location of the face center. Similar to calculating the actual voting of the hypothesis, the absolute voting to the face center in return is $y_i + \Delta_l^{zc}$, which is the actual form that is accumulated into the Hough space. The voting function is calculated like in Eq. 5. Thus the score function is:

$$S(z) = \sum_i v(z|I_i) = \sum_i \sum_{h \in H} v(h, z|I_i) \tag{8}$$

Then a mean-shift [34] algorithm is employed on the Hough map to find the mode. This is used as an estimate of the latent variable, $\hat{z}$. We then define a region around $\hat{z}$ as

$$\phi(\hat{z}) = \{z : ||z - \hat{z}|| \leq r\} \tag{9}$$

The radius r is learned at training time. The sieve filters out the patches which cast votes out of this region, i.e., retains only the votes that are consistent with the estimate of the latent variable. The voting model for the hypothesis is the same as described in Eq. 5. The score function of hypothesis h after the latent continuous sieve can be written as:

$$S(h) = \sum_{i, z \in \phi(\hat{z})} v(h, z|I_i) \tag{10}$$

It shares the same form of Eq. 7 but with different $\phi(\hat{z})$ property since z here is in continuous space.

As shown in Fig. 2d, after filtering by the sieve, voting elements that violate the face center consistency and vote for other face center hypotheses, are removed from the votes set. The ones that satisfy the face center consistency are kept.

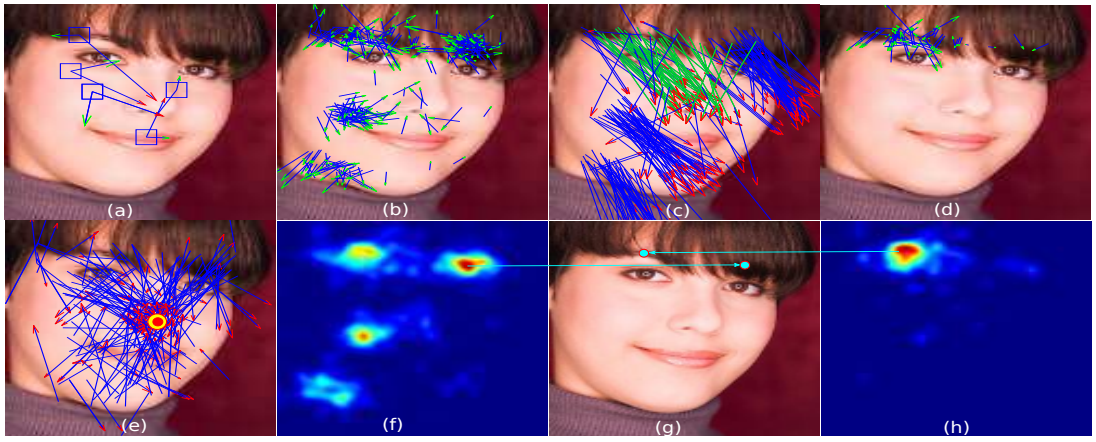


Fig. 2: Illustration of sieving via continuous latent variable (face center). (a) A voting element consists of two offset vectors, one to the target point (green arrow) and the other to face center (red arrow). (b) Original set of votes for the left brow center. (c) The absolute face center votes, those in green are regarded as consistent to the face center. (d) The remaining voting elements filtered by the face center sieve. (e) All voting elements are used to localize the face center (red dot). (f) and (h) are the Hough maps generated from votes of (b) and (d) respectively. (g) shows the corresponding detection results.

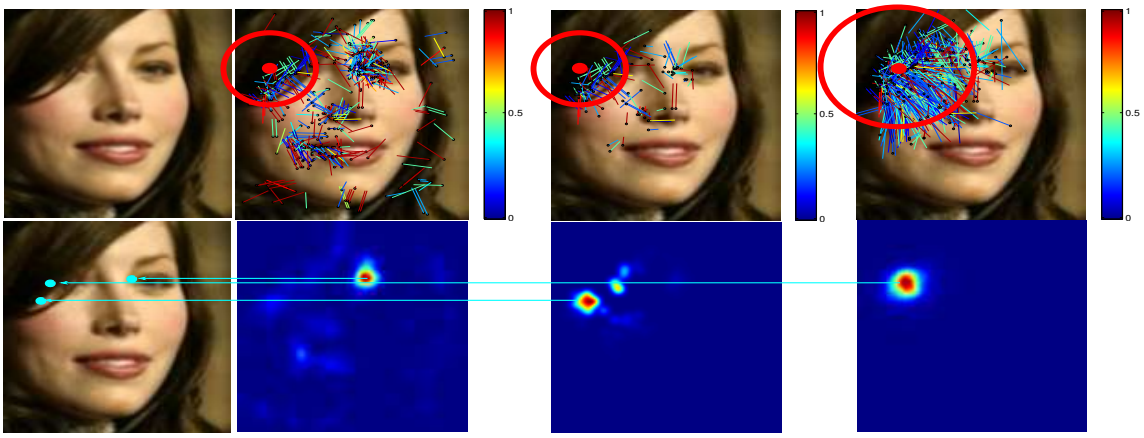


Fig. 3: Illustration of aggregating the votes by updating the threshold. From left to right, the first row shows the original face image, all votes for the point ($\lambda = 0.35$), votes passed face center sieve and the aggregated votes from updated threshold ($\lambda = 0.22$) passed face center sieve. The color represents the weight of each vote and the dark terminal is the voting destination. The second row shows the detection results, normalized Hough map for original voting, after face center sieving and re-voting.

C. RF Votes Aggregating

Taking all the voting elements into account for each hypothesis can lead to bias towards the mean shape and also it is very time consuming. Thus in practice, when collecting the votes for an individual feature point, a threshold is applied, similar to [3], [11], [9]. This works as a filter that prohibits votes with large offsets. This threshold is typically optimized during training and kept fixed during testing. Only the votes that satisfy threshold are allowed to vote for the hypothesis, i.e., $f(\Delta) > \lambda$, where $f(\cdot)$ is the proximity function defined in Eq. (2).

This mechanism works well in most cases but fails, for example when a feature point is heavily occluded. As shown in Fig. 3, in the presence of a heavy occlusion, only few valid voting elements remain after the face center sieve is applied. This is expected since in the case of heavy occlusion, there are no patches near the occluded facial landmark that can cast reliable votes. In such cases, we should allow votes from patches that are farther away. Thereby we need to reduce the proximity threshold. Such votes, introduce implicit facial shape constrains. In order to determine an image-dependent proximity threshold λ_j for the j-th landmark, we pose it as an rare event detection problem using one class SVM (OC-SVM) [35], which, given a certain value λ_j and the voting map calculated using that threshold, determines whether the threshold should be decreased or not. In order to train an OC-SVM for each facial point, we collect a set of positive training instances, i.e., the images in which the facial point is not occluded and can be localized accurately using the current proximity threshold. We propose to use middle-level features that are extracted directly from the votes set after the object center sieve is applied. The feature is represented as a histogram of the voting orientation. Specifically, we first compute the voting center using a mean-shift algorithm, then the votes are divided into four separated sub-windows using the x-y coordinate system originated at the voting center. Then

we calculate the voting orientation histogram in each sub-window, with 12 equally divided bins, i.e., 30° per bin. This results in a 48-dimensional feature denoted by x_1. As shown in Fig. 4, the histogram of voting orientation of occluded facial points (the right one) differ significantly from non-occluded ones (the left two). By contrast, the histograms of non-occluded landmarks are similar, despite the fact that the face images are quite different. Given the features of positive training instances for each facial point, a RBF-kernel based OC-SVM model is learned that is able to determine whether or not to adjust the proximity threshold.

We also calculate another feature, that is the ratio of votes after and before the face center sieve is applied, $x_2 = \frac{|V^F|}{|V|}$. V and V^F are respectively the set of votes before and after the face center sieve is applied. If x_2 is less than a threshold τ, then the proximity threshold should be reduced. In order to determine how much the proximity threshold should be reduced for a certain facial landmarks, we consider whether our classification scheme has determined that the threshold for neighbouring landmarks should be reduced or not as well. This is an indicator that the corresponding patches around them are also unreliable (e.g., there is occlusion). Therefore the proximity threshold reduction should be larger. We define two neighbours $j' \in Ne(j)$ for each landmark. The votes aggregating, or proximity threshold updating procedure is summarized in Algorithm 1.

Algorithm 1 Aggregating votes

Input: $\Lambda = \{\lambda_j\}$ with pre-optimized proximity thresholds
Output: Updated proximity thresholds Λ

initialize the update index vector $K = \{k_1, ..., k_j, .., k_J\}$ with all zeros ▷ # of steps to update
for all $j \in \{1, ..., J\}$ **do**
  collect voting elements V_j based on λ_j
  apply face center sieve and obtain V_j^F
  calculate the middle level feature x_1 and x_2
  $Rt \leftarrow \text{svm}_j(x_1)$ ▷ apply the OC-SVM
  if $Rt == -1$ **or** $x_2 < \tau$ **then** ▷ τ,threshold
    $k_j := k_j + 1$
  end if
end for
for all $j \in \{1, ..., J\}$ **do**
  for all $j' \in Ne(j)$ **do**
    if $k_{j'} > 0$ **then**
      $k_j := k_j + 1$
    end if
  end for
  $\lambda_j := \lambda_j - k_j * step * \lambda_j$ ▷ *step*=0.3
end for

D. Landmark Unreliability

Face analysis systems, for instance face recognition and facial expression, suffer a lot from the partial occlusion caused by hair, hand, sunglasses, scarf or other objects. [36] has studied the face recognition performance drop due to partial occlusion. In general, occlusions can lead to two main problems. First it leads errors in the object alignment and second it leads to the extraction of features from places that do not contain facial information. Most of alignment methods give out the result as is. The subsequent feature extraction step can only assume both 100% correctness of the alignment and equal importance of the features from different landmarks. In our method, we give out the unreliability of each landmark, in a way of discrete levels ranging from 0 to 3, 4 levels in total, which shares the value of threshold updating index k_j calculated from Algorithm in 1. Larger value mean higher unreliability and landmarks with $k_j = 0$ are reliable. Note our method compensates for the cases where nearby votes are no reliable (i.e., $k_j > 0$). $k_j > 0$ does not necessarily mean occlusion is presented. We argue that even though the points under heavy occlusion can often be localized in a high accuracy, the features extracted from the nearby region are not reliable for further object analysis, such as face recognition or car reconstruction.

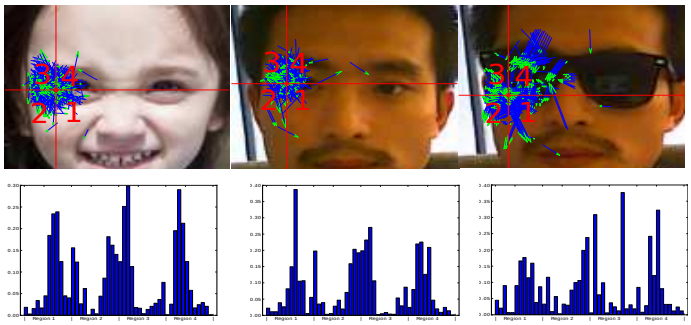


Fig. 4: Feature extracted from the votes passed the face center sieve. The left shows an example image for training the one class SVM classifier for the left eye corder. The middle shows an example tested positive and the right shows an example tested as outlier. The red lines split the votes into four regions and the below shows their corresponding features, i.e., x_1.

IV. Experiments

To evaluate the efficacy of the proposed method, we conduct experiments on two face databases and one car database, both collected in uncontrolled environments. Object alignment in both cases is challenging since a) most car/face landmarks are only weakly discriminative for detection; b) the images are taken from various viewpoints and c) often contain cluttered backgrounds and severe partial occlusion.

A. Database Description

Labelled Face in the Wild (LFW) [37] contains 13,233 face images, annotated in [3] with the location of 10 facial points and the face bounding boxes. The variety of images in this dataset is less challenging but all the images are of low quality.

Annotated Face Landmarks in the Wild (AFLW) [38] contains real-world face images from Flickr. These images exhibit a very large variability in pose, lighting, expression,

ethnicity, hairstyles as well as general imaging and environmental conditions. Many images exhibit partial occlusions that are caused by head pose, objects (e.g., glasses, scarf, mask), body parts (hair, hands) and shadows. In total, 25993 faces are annotated with up to 21 landmarks per face. We selected a subset in which all 19 frontal landmarks (i.e. excluding the two ear lobes) were annotated (about 6200 images). From this subset, we randomly select 1000 images for testing and 600 images for validation (300 of which were selected manually to ensure they contain occlusions). The rest were used for training the forest.

CMU Cars in Wild (CMU-CW) [14] contains 3433 cars spanning a wide variety of types, sizes, backgrounds and lighting conditions including partial occlusions. The images are from MIT Street Dataset [39] created for the task of object recognition and scene understanding. The car landmarks were annotated in [14]. The labelled data was manually classified into five different views: 932 frontal view, 1400 half-front view, 1230 half-back view and 1162 back view images. The car shape is respectively represented by 8, 14, 10, 14 and 8 landmarks.

Throughout our experiments we report the root mean square error (RMSE) of the localization of the landmarks with respect to the manually labelled ground truth landmark locations. The error in the face images is normalized as a fraction of the inter-ocular distance as in [3], [6], [16].

B. Implementation Details

Forest Model on the LFW We use the trained model from [3], denoted by **CRF** in this paper, on the LFW as a baseline for comparison. At each leaf node, the trained model provides offset vectors to 10 facial points and it also provides a mean patch offset vector to the center of the bounding box. The latter is treated as a continuous latent variable for sieving the votes, i.e. Δ_l^{cz} in our work. We assign a unit weight to each vote, i.e. we set $\omega_l^{cz} = 1$. We denote this forest model with the bounding box center as continuous latent variable by **CRF_C**. This allows us to evaluate our contributions using the CRF as a baseline.

Forest Model on the AFLW We show the contribution of each component of our method on AFLW by training models that are listed in Table I. The trees in the forests F1, as in [3] are trained without using any additional information. In order to train forests with sieves using latent discrete variable, i.e. F2-F5, we quantize the head yaw angles of the training samples into 3 labels like [3]. We train a forest using the additional discrete information to learn multiple voting models at leaf nodes as described in Section III-B1. A similar idea is proposed by Sun *et al.* [9] for human pose estimation. We denote their method by CRF-S. In forests F3, F4 and F5, each vote at leaf node contains voting information to the face center as described in Section III-B2. In the forest model of F4, we set the proximity threshold of individual facial point to 0, i.e. allow all the votes from the face to vote for the facial point. The tree model of F5 is the same as F4 but performs threshold adjustment as described in Section III-C. We use the same macro settings of the forests of [3] such as the image features, maximum tree depth (20), number of tests at the internal nodes (2500), forest size (10 trees in total) and the bandwidth of the mean-shift algorithm. Also we use the same random subset of the training samples for the same index of tree in each forest in order to avoid a bias caused by random sampling of the training data.

TABLE I: Description of forest models trained on the AFLW

Forest ID	Sieves		Aggregation
	Discrete	Continuous	
F1	No	No	No
F2 (CRF-S)	Yes	No	No
F3	Yes	Yes	No
F4	Yes	Yes	Max. aggregating
F5	Yes	Yes	Yes

Forest Model on the CMU-CW We train one forest for each of the 5 views using the training set-up used in [14]. We randomly select 400 images for each view for training and use the rest for testing. We sample 30 patches sized 30×30 from a non-occluded landmark region for training. We use the car center, calculated as the mean value of all the landmarks, as a continuous latent variable in this model. A tree in the forest is trained on 300 randomly sampled car images and 4 trees in total are trained for each view.

Parameters for votes sieving The key parameter associated with the continuous variable sieve, that is the radius r is set to 0.3 through a grid search on the AFLW validation set. We use the same sieving parameters for LFW and CMU-CW.

Parameters for votes aggregating For each facial landmark, we select the most accurate 500 detections (localization error less than 0.1) from the AFLW validation dataset as positive training samples to train the OC-SVM model. When there are not enough training samples we select some from the training samples. The OC-SVM models of the CMU-CW is directly trained on the training samples. We use the LibSVM [40] to train the OC-SVM model.

C. Method Evaluation

In this section we evaluate the influence of the different components of our models and summarize our findings from the experiments performed on the AFLW dataset. We repeat the experiment for 4 times. The reported results below are averages over the 4 runs.

1) Performance of votes sieving: Since the sieving can be based on both discrete and continuous latent variables, we evaluate them separately.

Sieving via discrete latent variable. In order to evaluate the efficacy of sieving via discrete latent variable, i.e. the discrete head pose in our case, we report the results using forests F1 and F2. As can be seen from Fig. 5 where the cumulative distribution of facial points over error threshold is shown, the forest with sieves associated with the discrete head pose label performs significantly better. However, neither of them is able to deal with challenges like occlusion and shadows, and only a proportion of the facial points can be localized with high accuracy. The percentages of facial points with error less than 0.1 are respectively 65% (F1) and 70% (F2).

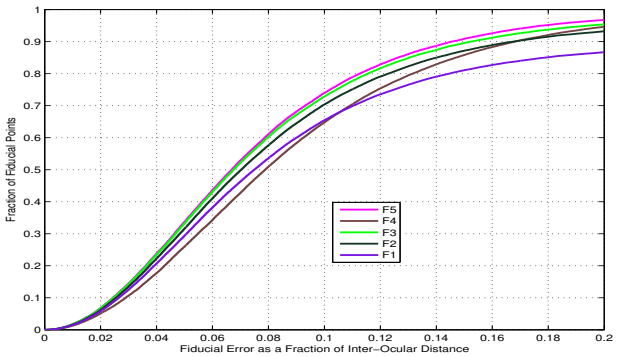


Fig. 5: Error distribution.

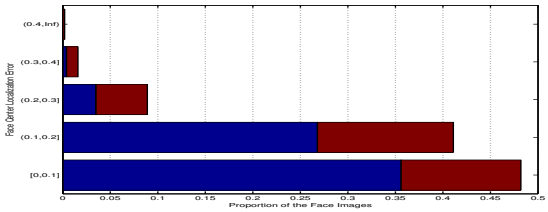


Fig. 6: Face center estimation error distribution.

Sieving via continuous latent variable. Since sieving using continuous latent variables involves estimating the latent variable, we first we evaluate the stability of the estimation by measuring the error, in our case, the face center localization error. As is shown in Fig. 6, though the localization is not highly accurate, the performance is very stable: only 2 out of the 1000 test images have localization error larger than 0.4 and more than 98% of them have localization error less than 0.3. We note that accurate localization is not needed/done by the center sieve since we do not use an explicit shape model centred around it.

We compare the results using F2 and F3 in terms of localization error of each individual facial point. The relative improvement of F3 in comparison to F2, that is defined as the error reduction over the original error, is shown in Fig. 7. There are four facial landmarks (the two eye brow and eye outer corners) with more than 40% relative improvement over the baseline (F2) in mean localization error. Three facial points (right eye left corner, nose left and nose right) show less than 10% relative improvement since these points are less frequently occluded and therefore easier to localize. In order to illustrate better the efficacy of the sieves on difficult images, we split the test set containing 1000 images into two sets, **AFLW_TestI** and **AFLW_TestII**, the former containing

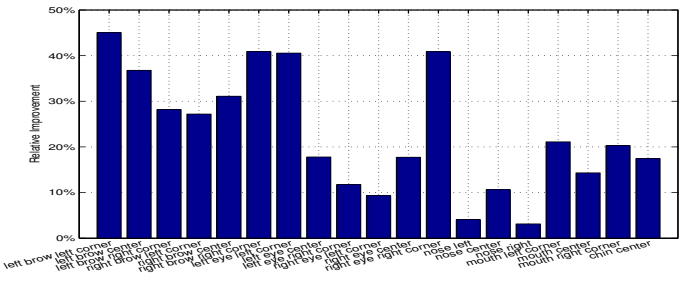


Fig. 7: Relative improvement by using the face center sieve.

"easy" and the latter containing "difficult" images. We do so by applying the F2 detector (now regarded as a baseline) on the whole test dataset and putting into the AFLW_TestI the face images with average localization error less than 0.1 (663 face images on average) and into the AFLW_TestII the rest (337 face images on average). We report results on them separately. As shown in Fig. 8a, in the easy set, applying the sieve only has very minor improvement, 2.3% in average. By contrast, in the difficult set, as shown in Fig. 8b, the improvement is very significant. The average relative improvement of the 19 points is 38% and the improvement is more significant for the difficult facial points, for instance the left eye brow left corner (51.5%, 0.1404 vs 0.2895), the right eye right corner (62.3% 0.0910 vs. 0.2413). The superior performance of F3 over F2 significantly validates the efficacy of our sieves, particularly on 'difficult' images.

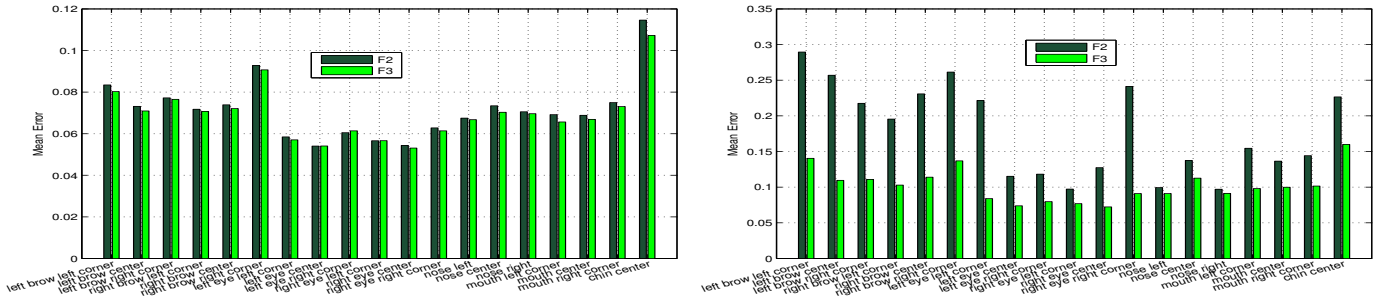
2) Performance of votes aggregating: The aggregating of the votes is controlled by a proximity threshold associated with individual facial landmark. In F3 we use threshold that are optimized for each facial landmark during training and in F4 reset the proximity threshold to infinity , that allows votes from the whole face region. The results are shown in Fig. 5. By taking all votes into account, F4 has the lowest performance for errors less than 0.1. Its distribution rises to a similar level to F3 and becomes higher than F2 at around 0.2 error. This shows that taking all votes into account leads to robustness but degrades the localization accuracy. The efficacy of votes aggregating is best shown by comparing the results of F5 and that of F3. Even though we cannot observe large margin of improvement in this figure, we note that the votes aggregating performed only when it is necessary, in most cases when heavy occlusion is present. In our four test experiments, the proportion of facial points with different aggregating steps (defined in Algorithm 1) is shown in Table II, in total only 20% of them adjust the threshold to aggregate the votes.

TABLE II: Aggregating steps proportion

Steps	$k=1$	$k=2$	$k=3$	Total
Percentage	10.5%	7.51%	2.43%	20.44%

We also evaluate the overall performance of using the sieves and aggregating by comparing F5 with F2. Though F2 has better performance than the plain forest F1 and is formalized as a type of our sieves, we treat it as the baseline method here because we want to highlight the original contribution of this work, as the idea of F2 originally proposed in [9] for human pose estimation. The improvement plot over the baseline error (CRF-S) is shown in Fig. 9, which validates that the improvement is correlated with the 'difficulty' level of the test images, i.e., our method produces large improvement when the baseline method has big error.

3) Landmark unreliability: We qualitatively show some examples of facial landmarks unreliability detection in Fig. 10, where the number associated with each point location, that is also the aggregating step, intuitively reflect the unreliability level of the point. Since the unreliability of a region can be caused by several reasons, it is very difficult to determine it using low-level image feature. Our method model explores



(a) (b)

Fig. 8: Performance of sieves associated with the face center on the AFLW. The left and right are landmark-wise mean error results on AFLW_TestI and AFLW_TestII respectively. Note that the Y axis range of (b) is different from that of (a).

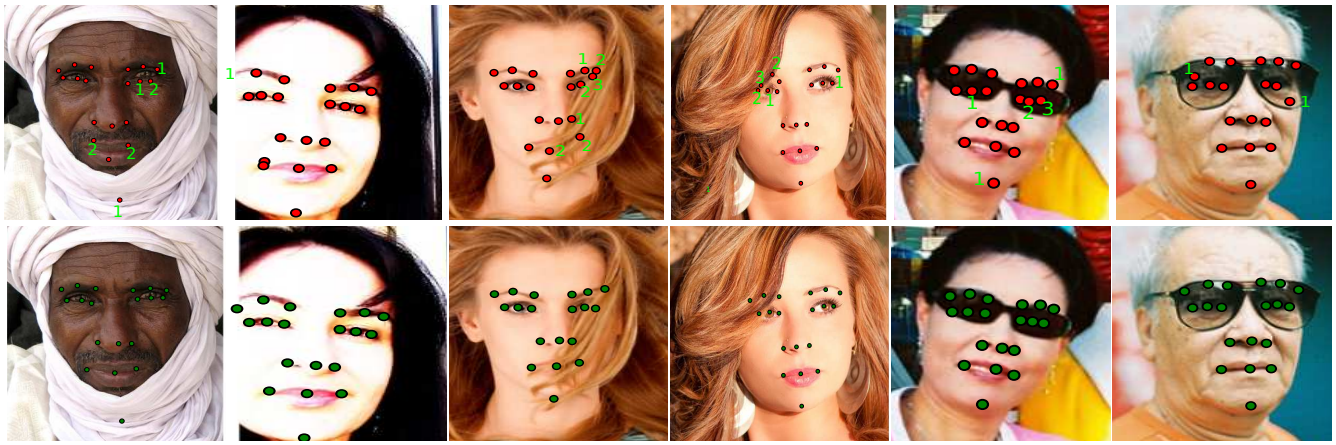


Fig. 10: Example results from the AFLW dataset before (top row) and after (bottom row) the votes aggregating. The value beside the red dot in the top row indicates the unreliability/ step length of aggregating. For clarity, the reliable point where no aggregating is needed, i.e. 0 is not shown in the figure.

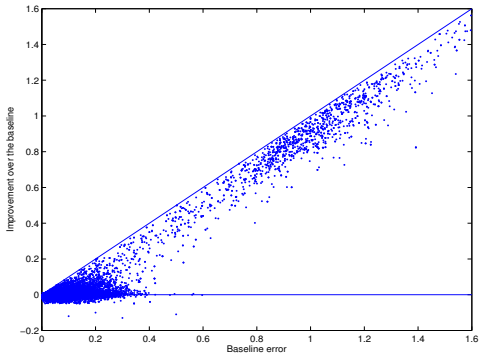


Fig. 9: Improvement plot over the baseline error (CRF-S).

the information from middle level features, extracted from the voting map. We also note that some unreliable facial points, like the eye corners under sunglasses in the last two columns, are not well identified. This is because during when training time of the OC-SVM, such images are used as positive training samples since the localization accuracy is high. Thus further validates our sieving step is very robust to such kind of occlusion.

D. Face Alignment Comparison

In this section we compare the performance of our proposed method with the existing face alignment approaches, namely the closely related random forests-based methods and other state-of-the-art methods. We do so on several widely used datasets.

Comparison with [3] on LFW A work that is closely related to our method is the CRF proposed in [3]. It reports the best performance on the LFW dataset. We evaluate the contribution of our sieve associated with latent continuous variable by comparing with its publicly available trained model. We randomly select 1000 images from the dataset for testing and split them into two sets, namely **LFW_TestI** and **LFW_TestII** according to the average localization error of the CRF detector. In this way we create an 'easy' partition, namely the **LFW_TestI**, where the average point localization error of the CRF is less than 0.1, and a 'difficult' partition, namely the **LFW_TestII**, where the average point localization error of the CRF is larger than 0.1. We repeated this 4 times and on average 118 out of 1000 face images ended up into LFW_TestII. This small number is due to the fact that the face images in the LFW dataset are relatively easy. Only a few of them contain occlusions caused by head pose, hair and sunglasses. The absolute improvement of mean error and

Fig. 11: Detection results of example images from LFW. The upper shows the results by CRF detector [3] and the lower shows the results of our method.

accuracy (using the definition of [3]) on the LFW_TestI and LFW_TestII are shown in Fig. 12 and Fig. 13. On LFW_TestI, there are some points our method even performs slightly worse, but the difference is negligible. To give the reader an idea, the maximum difference in the average point error is around 0.05 pixels. The maximum difference in the accuracy is also very small, namely around 0.5%. This is expected since our method is designed to maintain the performance of the baseline regression forests on "easy" images. On the contrary, the improvement on LFW_TestII is noticeable. The absolute reduction in the mean error for the *left eye left* point in average is around 0.4 pixels and that of the *right eye right point* is around 0.3 pixels. The differences on other points are not so noticeable. There are three points (*left eye left*, *left eye right* and *right eye right*) with more than 6% increase in detection accuracy.

As can be seen from the example images shown in Fig. 11, since the CRF detector [3] localizes each individual landmark in a completely independent way, there are some points that are localized incorrectly due to occlusion or shadows caused by pose, hair or glasses. On the contrary, after applying our sieves associated with the face box center, based on the same trained model, our method is able to deal with the partial occlusion in an efficient way.

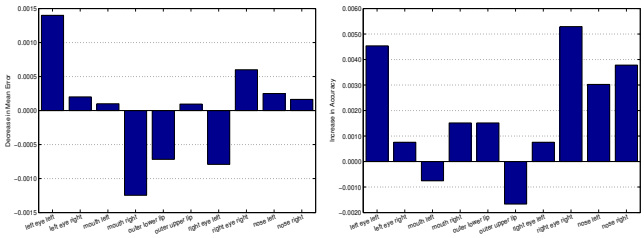


Fig. 12: Results on the LFW, compared to [3]. The left and right are respectively the mean error decrease and accuracy increase on the LFW_TestI.

Comparison on AFLW We compare the overall performance of our proposed method with methods from the academic community as well as commercial systems, namely (1) the structured-output regression forests (SO-RF) in [20], (2) the regression forests based CLM (RF-CLM) [41], (3) the mixture-of-trees (Mix.Tree) [5], (4) Xiong and De la Torre's Supervised Descent Method (SDM) [25] and (5) betaface.com's face detection module [42]. Since betaface.com, Mix.Tree models and SDM detector embed face detection with landmarks detection, for fair comparison we build our algorithm on top of a Viola-Jones face detector from the Matlab computer vision toolbox. We manually discard missed or incorrect detections (e.g. sometimes Mix.Tree detected a half face) by any method when calculating the error. Among 1000 images, there are 74 missed face detections for betaface.com, 113 for Mix.Tree, 127 for SDM, and 89 for Matlab Viola-Jones detector. Though SDM also uses the Viola-Jones face detector [43] in OpenCV, the result is slightly worse than that provided Matlab toolbox, probably because different trained models are applied. Mix.Tree failed to detect small faces because they were trained on large faces where all landmarks are clearly visible. The test set then contains 776 images (555 in AFLW_TestI and 221 in AFLW_TestII). We compare results of 11 common points to CRF-S, betaface.com, SDM and Mix.Tree as shown in Fig. 14a and Fig. 14b. On the AFLW_TestI we see that both CRF-S and our method perform better than Mix.Tree and betaface.com, and slightly worse than SDM. On AFLW_TestII, CRF-S performs significantly worse while the other existing methods and our method have more stable performance. Our method performs better than Mix.Tree and betaface.com, and on par with SDM.

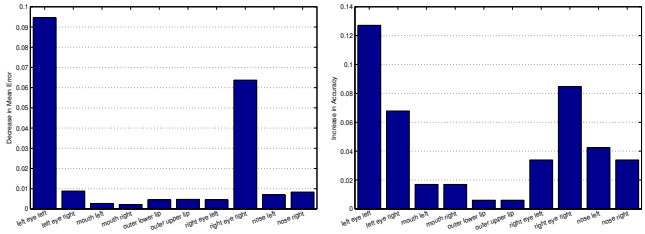


Fig. 13: Results on the LFW, compared to [3].The left and right are respectively the mean error decrease and accuracy increase on LFW_TestII Note that the range of the Y axis is different from that of Fig. 12.

In Fig. 15 we compare the average localization error of all the 17 internal points on a face (the chin center and mouth center are excluded) of our method with the random forests-based method, i.e., CRF-S, SO-RF and RF-CLM. We train SO-

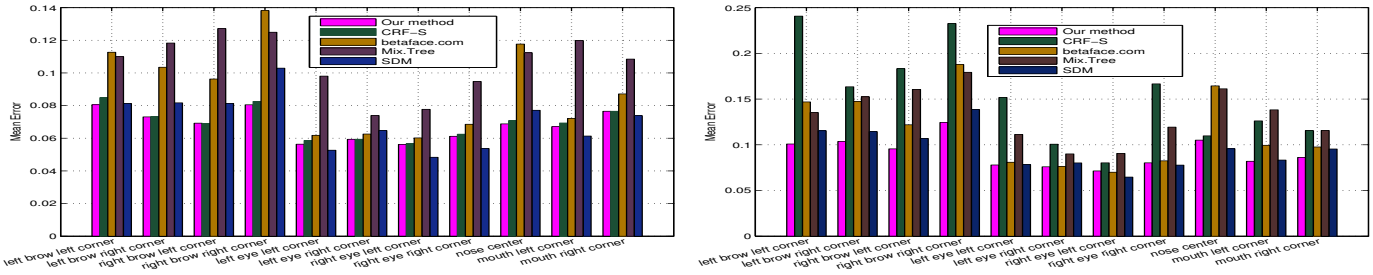


Fig. 14: Results of our method on the on AFLW_TestI (Left) and AFLW_TestII (Right), compared to [9], [25], [5] and betaface.com [42].

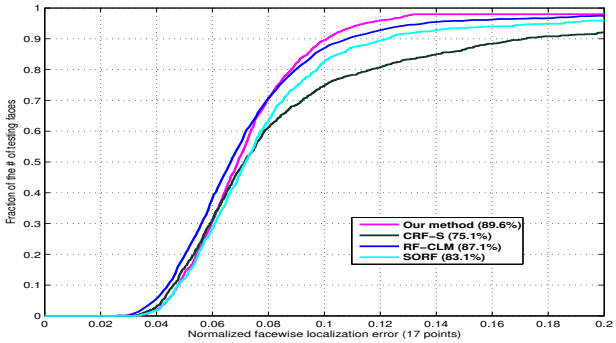


Fig. 15: Results of our method on the AFLW, compared to random forests-based methods [20], [41]. The numbers in legend of (c) are the percentage of test faces that have average error below 10%.

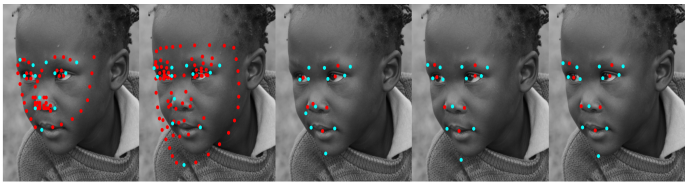


Fig. 16: Left to right: Results for Mix.Tree, betaface.com, CRF-S, SO-RF and our method on an image from AFLW. The blue dots are the 12 common points.

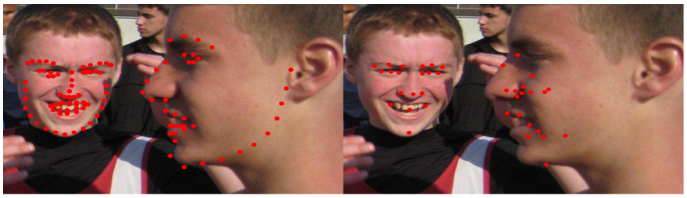


Fig. 17: An example image from AFW [5] with results from Mix.Tree (Left) and our method (Right)

RF model on AFLW using the code provided by the authors using the same experimental setting as that of CRF-S and compare the reported result of RF-CLM. The markup of RF-CLM is slightly different since their results are for 17 points but two are annotated by the authors, that are not publicly available. As can be seen in Fig. 15, where the error cumulative distribution of the random forests-related methods is shown, our method performs on par with RF-CLM and significantly better than SORF and the CRF-S, though both RF-CLM and SORF are based on shape model fitting. An example image is shown in Fig. 16 where our method performs better than not only the local detection method, like CR-S, but also the ones using shape models such as [5], [20]. In addition, we have found that in terms of computational complexity and in terms of how well it deals with low quality images, our method performs considerably better than the Mix.Tree model. However, as shown in Fig. 17, unlike the Mix.Tree, our method fails on side view faces since we have not used such images in training.

E. Car Alignment Comparison

We evaluate our method on car alignment using the same experimental set-up presented in [14]. More specifically, for each view, the landmark-wise average RMSE over four different subsets is reported. More precisely, 1) the average over all images, 2) the average over images with occluded landmarks, 3) the average over the unoccluded landmarks in partially occluded images and 4) the average over the occluded landmarks in partially occluded images. The results are shown Fig. 18. From top to bottom are the results of the four different subsets respectively and from left to right are the results for views from 2 to 4. The front and back view images are less challenging and their results are not shown here.

We compare the baseline regression forests and two other methods, the Random Forests (RFs) based method proposed by Li et al. [7] and the Vector Correlation Filter (VCF) method by Boddeti et al. [14]. We compare with their best reported results [14], i.e. the results from RFs with RANSAC BPSI shape model and VCF with Greedy BPSI shape model (see [14] better for details). We observe that our method is able to align most of the landmarks in lower RMSE for different subsets of view2 and view3. For view4, our method performs better than the RFs-based method and on par with the VCF method. To further investigate the error distribution we also compare the individually sorted errors for each view in Fig. 19. We observe that in view2 and view3, our method performs significantly better, i.e., for a given error tolerance our method aligns more images compared to state-of-the-art methods while the baseline regression forests-based method performs worse. In view4, our method performs better than RFs and similar to VCFs. The superior performance over the baseline plain regression forests validate the efficacy of our proposed votes sieving and automatic aggregating. Example results from all

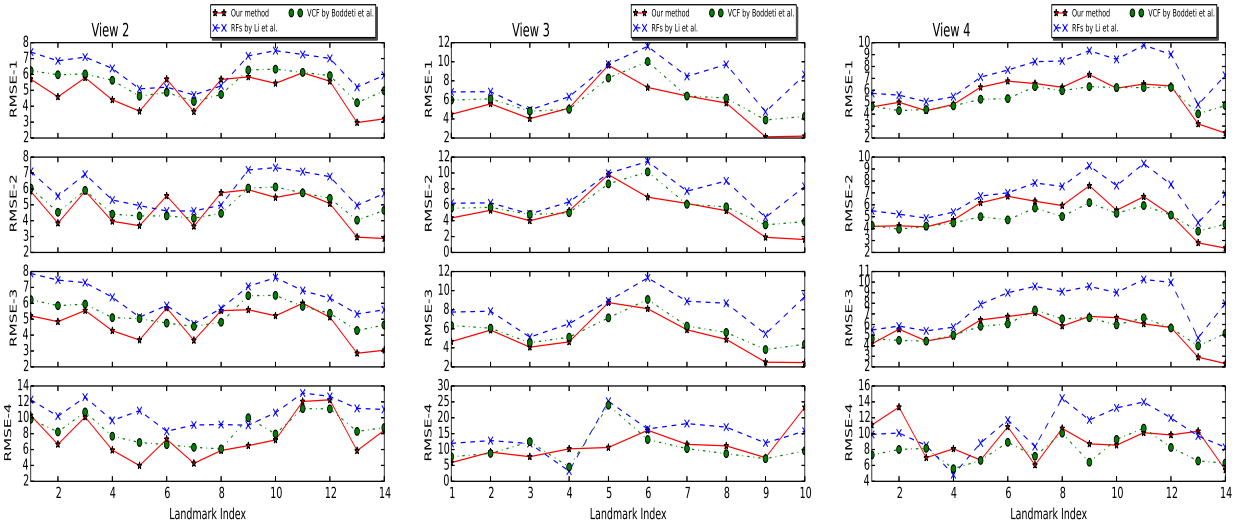


Fig. 18: Landmark wise RMSE error for each view, from top to bottom: 1) all image, 2) images with no occlusions, 3) unoccluded landmarks of partially occluded image, 4) occluded landmarks of partially occluded image.

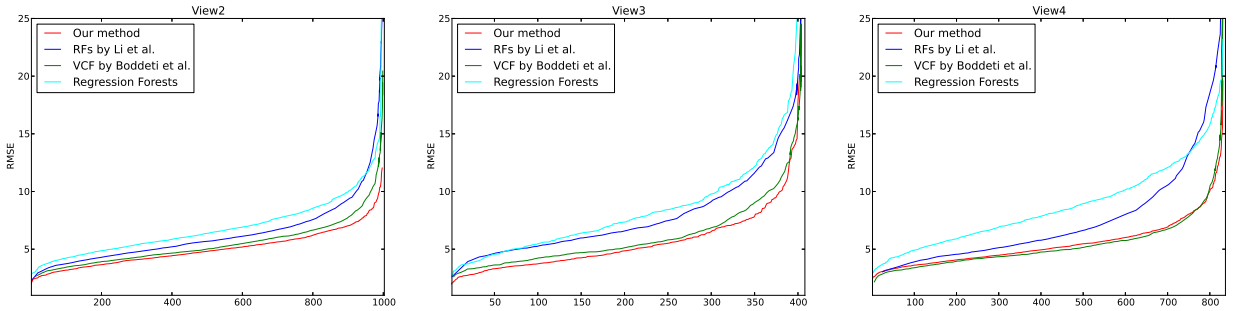


Fig. 19: Comparison of the sorted RMSE for each view to the VCF model in [14], random forests model in [7] and the baseline Regression Forests in our work.

.

the five views are shown in Fig 20. The top row shows the results from the plain regression forests, that is unable to handle occlusions. The bottom row shows the results of our method.

V. Conclusion

This paper presents a regression forests votes refining method for object alignment problem. Before accumulating the votes to a Hough map for detection, it filters out the false positives votes by using sieves which impose agreement on latent discrete or continuous variables. In addition, it proposes a votes aggregating strategy which automatically seeks additional votes when necessary. Our proposed method is validated on two challenging tasks: facial feature detection and car alignment. It yields performance superior or close to the state-of-the-art on the most challenging datasets with images collected in the wild. Our results raise some interesting questions. Other than the face center consistency, can we develop more latent variable sieves to filter out irrelevant votes before accumulating them into the Hough map? Can we extract more useful middle-level features from the votes for high-level vision tasks such as to measure the object similarity or to recognize the facial expression? Also, the proposed strategy can be naturally applied to other applications such as body joint localization. We plan to investigate these questions in our future work.

Acknowledgment

This work is partially supported by EU funded IP project REVERIE (FP-287723). Heng Yang is supported by a CSC/QMUL joint PhD student scholarship.

References

[1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011.

[3] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[4] B. Efraty, C. Huang, S. K. Shah, and I. A. Kakadiaris, "Facial landmark detection in uncontrolled conditions," in *Proc. Int'l Joint Conference on Biometrics*, 2011.

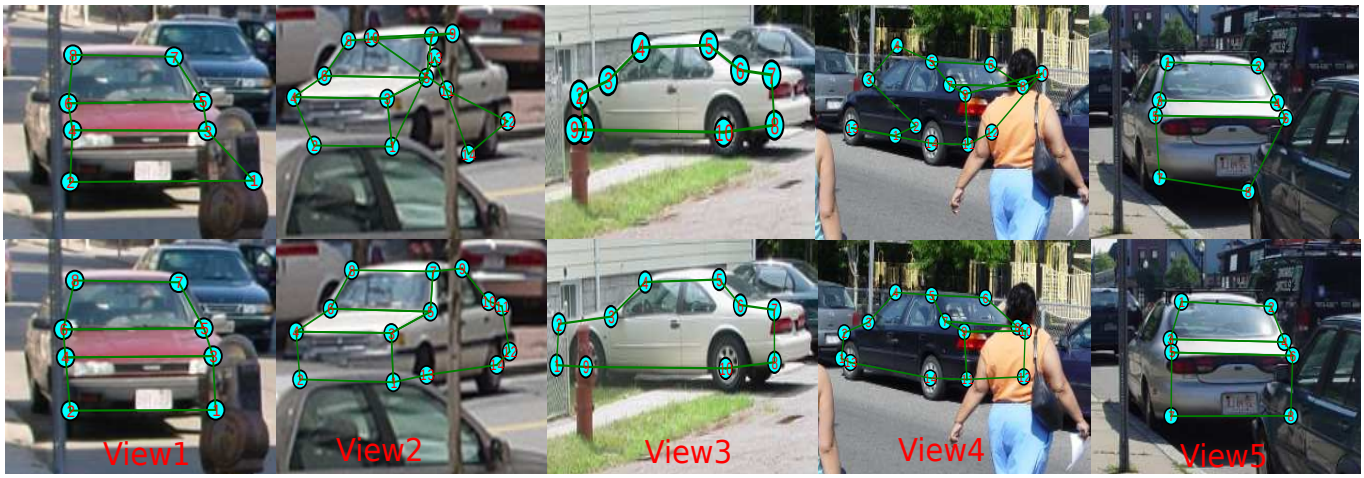


Fig. 20: Detection results of example images of different views from CMU-CW. The upper shows the results by plain regression forests and the lower shows the results of our method.

[5] X. Zhu and D. Ramanan, "Face detection, pose estimation and landmark localization in the wild," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[6] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[7] Y. Li, L. Gu, and T. Kanade, "Robustly aligning a shape model and its application to car alignment of unknown pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1860–1876, 2011.

[8] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[9] M. Sun, P. Kohli, and J. Shotton, "Conditional regression forests for human pose estimation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[10] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011.

[11] P. Kontschieder, S. R. Bulò, M. Donoser, M. Pelillo, and H. Bischof, "Evolutionary hough games for coherent object detection," *Computer Vision and Image Understanding*, 2012.

[12] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using Gabor feature based boosted classifiers," in *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics*, 2005.

[13] C. T. Liao, Y. K. Wu, and S. H. Lai, "Locating facial feature points using support vector machines," in *International Workshop on Cellular Neural Networks and Their Applications*, 2005.

[14] V. N. Boddeti, T. Kanade, and B. V. Kumar, "Correlation filters for object alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013.

[15] D. Cristinacce and T. Cootes, "Feature detection and tracking with constrained local models," in *Proc. British Machine Vision Conference*, 2006.

[16] B. Martinez, M. Valstar, X. Binefa, and M. Pantic, "Local Evidence Aggregation for Regression Based Facial Point Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

[17] B. Amberg and T. Vetter, "Optimal landmark detection using shape models and branch and bound," in *Proc. IEEE Int'l Conf. Computer Vision*, 2011.

[18] J. M. Saragih, S. Lucey, and J. F. Cohn, "Face alignment through subspace constrained mean-shifts," in *Proc. IEEE Int'l Conf. Computer Vision*, 2009.

[19] F. Zhou, J. Brandt, and Z. Lin, "Exemplar-based graph matching for robust facial landmark localization," in *Proc. IEEE Int'l Conf. Computer Vision*, 2013.

[20] H. Yang and I. Patras, "Face parts localization using structured-output regression forests," in *Proc. Asian Conf. Computer Vision*, 2012.

[21] ——, "Sieving regression forests votes for facial feature detection in the wild," in *Proc. Int'l Conf. Computer Vision*, 2013.

[22] J. Saragih and R. Goecke, "A nonlinear discriminative approach to aam fitting," in *Proc. IEEE Conf. Computer Vision*, 2007.

[23] P. A. Tresadern, P. Sauer, and T. F. Cootes, "Additive update predictors in active appearance models." in *Proc. British Machine Vision Conference*, 2010.

[24] P. Dollar, P. Welinder, and P. Perona, "Cascaded pose regression," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.

[25] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. IEEE Int'l Conf. Computer Vision*, 2011.

[29] A. Criminisi, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2011.

[30] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *Int'l J. of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[31] O. Barinova, V. S. Lempitsky, and P. Kohli, "On detection of multiple object instances using hough transforms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1773–1784, 2012.

[32] I. Patras and E. Hancock, "Coupled prediction classification for robust visual tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1553–1567, 2010.

[33] N. Razavi, J. Gall, P. Kohli, and L. Van Gool, "Latent hough transform for object detection," in *Proc. European Conf. Computer Vision*. Springer, 2012.

[34] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[35] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval," in *Proc. Int'l Conf. on Image Processing*, 2001.

[36] H. Ekenel and R. Stiefelhagen, "Why is facial occlusion a challenging problem?" *Advances in Biometrics*, pp. 299–308, 2009.

[37] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, Oct. 2007.

[38] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization," in *Proc. IEEE Int'l Conf. Computer Vision Workshops*, 2011, pp. 2144–2151.

[39] http://cbcl.mit.edu/software-datasets/streetscenes/.

[40] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
[41] T. F. Cootes, M. C. Ionita, and S. P., "Robust and Accurate Shape Model Fitting using Random Forest Regression Voting," in *Proc. European Conf. Computer Vision*, 2012.
[42] http://www.betaface.com/.
[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

Heng Yang (S'11) received the B.E. degree in Simulation Engineering and the M.Sc. degree in Artificial Intelligence and Pattern Recognition from National University of Defense Technology (NUDT), China, in 2009 and 2011 respectively. He is currently pursuing the Ph.D. degree in the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK since fall 2011. His research interests include computer vision and applied machine learning.

Ioannis (Yiannis) Patras (SM'11) received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Crete, Heraklion, Greece, in 1994 and 1997, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, Delft (TU Delft), The Netherlands, in 2001. He is a Senior Lecturer at the School of Electronic Engineering and Computer Science, Queen Mary University of London, U.K. His current research interests are in computer vision and pattern recognition, with emphasis on the analysis of human motion, including the detection, tracking, and understanding of facial and body gestures and their applications in multimedia data management, multimodal human computer interaction, and visual communication. He is an Associate Editor of the Image and Vision Computing Journal.